3M2-IOS-3b-4

Extracting Transliteration Pairs from Classical Chinese Buddhist Literature

Yu-Chun Wang*1 Richard Tzong-Han Tsai*2

*1 Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan

*2 Department of Computer Science and Engineering,
Yuan Ze University, Taiwan

Transliteration pair extraction, the identification of transliterations of foreign loanwords in literature, is a challenging key task in research fields such as historical linguistics and digital humanities. In this paper, we focus on one important type of historical literature: classical Chinese Buddhist texts. We propose an approach which can identify transliteration pairs automatically in classical Chinese texts. Our approach comprises two stages: transliteration extraction and transliteration pair identification. To extract more possible transliterations without introducing too many false positives, we adopt a hybrid method consisting of a suffix-array-based extraction step and a language-model based filtering process. Next, using the ALINE algorithm, we compare the extracted transliteration candidates for phonetic similarity based on their pronunciations in the middle Chinese rime book Guangyun. Pairs with similarity above a certain threshold are considered transliteration pairs. To evaluate our method, we constructed an evaluation set from several Buddhist texts such as the Samyuktagama and the Mahavibhasa, which were translated into Chinese in different eras. Precision and recall are used to measure and show the effectiveness of our method.

1. Introduction

Cognates and loanwords play important roles in the research of language origins and cultural interchange. Therefore, extracting plausible cognate or loanword pairs from historical literature is a key issues in historical linguistics. The adoption of loanwords from other languages is usually through transliteration. In Chinese historical literature, the characters used to transliterate the same loanword may vary because of different translation eras or different Chinese language/dialect preferences among translators. For example, in classical Chinese Buddhist scriptures, the translation process of Buddhist scriptures from Sanskrit to classical Chinese occurred mainly from the 1st century to 10th century. In these works, the same Sanskrit words may transliterate into different Chinese loanword forms. For instance, the surname of the Buddha, Gautama, is transliterated into several different forms such as "瞿曇" (qü-tan) or "喬答摩" (qiao-da-mo), and the name "Culapanthaka" has several different Chinese transliterations like "朱利槃特" (zhu-li-pan-te) and "周利槃陀伽" (zhou-li-pan-tuo-qie). Historians and Buddhist scholars currently use dictionaries to understand the relationships among different loanwords. However, it is difficult for any dictionary to cover all possible loanwords due to the vast amount of classical Chinese literature.

Identification of loanwords which are mostly transliterations is a key step in the construction of language families and language contacts. Methods employed in this step usually attempt to measure phonetic similarity among the plausible words. There are several approaches to extracting possible cognates based on phonetic similarity. Covington [Covington 96] proposed a cognate alignment algorithm that estimates phonetic similarity by finding the minimum-cost alignment through depth-first search. The cost of each alignment was measured in terms of the cost of substitution, insertion, or deletion of each segment. Kondrak [Kondrak 03] proposed the ALINE algorithm to measure the similarity between the phonemes based on the phonetic characteristics such as the place and the manner of articulation. He defined the salience and the value of each phonetic characteristic to measure the similarity of each phoneme pair. The actual similarity is the total sum of the similarity score of the phonemes on the optimal alignment.

In addition to the phonemic rule based methods, there are some methods based on machine learning approaches. Ristad and Yianilos [Ristad 98] adopted the expectation maximization (EM) to learn the probability of each string edit action and create a stochastic transducer to output the string edit distance of the two phonemic sequences. Mackay and Kondrak [Mackay 05] proposed a method to measure phonetic similarity based on a pair hidden Markov model (Pair HMM) which is often used in bioinformatics. The Pair HMM can generate the two output streams concurrently representing the alignment results of the two sequences. The Pair HMM has three states, which stand for the three based string edit actions: insertion, substitution, and deletion. They used the collected cognates as a training set to learn the parameters of the HMM model. The machine learning methods can improve the accuracy of word alignment and the phonetic similarity. However, the supervised learning method requires a labeled training set which is labor-intensive and not easy to obtain in many languages.

Contact: Richard Tzong-Han Tsai, Yuan Ze University, 135 Yuan-Tung Road, Chung-Li, Taiwan, (886)-3-4638800, thtsai@saturn.yzu.edu.tw

The method based on the phonemic rules can be easily construct deal with all the possible phonemic phenomena.

2. Method

Our loanword pair identification system comprises three components: term extraction, term filtering, and phonetic similarity estimation. An overview of our system is shown in figure 1.

2.1 Term extraction

In order to extract transliteration loanword from classical Chinese literature, we adopt the suffix array method [Manzini 04] to extract possible terms. Suffix array is a data structure designed for efficient searching of large bodies of text. The data structure is simply an array containing all indexes of the text suffixes sorted in lexicographical order. Each suffix is a string starting at a certain position in the text and ending at the end of the text. Searching a text can be performed by binary search using the suffix array. The term extraction procedure of suffix array is as follows: Given a string S with length n, for each character in the string S, create indexes from 0 to $n-1$, where $S[i]$ represents the suffixes of the index i. Supposing S = "abracadabra", after indexing, the index result would appear as follows:

Character	a	b	r	a	c	a	d	a	b	r	a
Index	0	1	2	3	4	5	6	7	8	9	10

The string S has 12 suffixes. We can sort these suffixes in the lexicographical order that is shown in the following table. The frequency is the count of this suffix, which appears as a prefix of other suffix strings.

Sorted Suffix	Suffix S	Frequency
a	$S[10]$	5
abra	$S[7]$	2
abracadabra	$S[0]$	1
adabra	$S[5]$	1
acadabra	$S[3]$	1
bra	$S[8]$	2
bracadabra	$S[1]$	1
cadabra	$S[4]$	1
dabra	$S[6]$	1
ra	$S[9]$	2
racadabra	$S[2]$	1

The suffixes whose frequency is greater than 1 are considered candidate terms. However, if a suffix is a substring of another suffix and its frequency is not greater than that suffix's, the suffix will be dropped. From the above example, we can extract two candidate terms "a" and "abra".

2.2 Term filtering

The suffix array method can extract a lot of possible terms. However, most of them are not actual transliteration terms. Thus, we have to filter out false positives before measuring phonetic similarity to avoid wasting computational resources and degrading the precision of pair identification. Sometimes the suffix array method may extract a term composed of an actual transliteration and one or more other words. For example, for the extracted candidate term "於憍薩羅國" (in Kosala state), if we want to separate out the actual transliteration, "憍薩羅" (Kosala), from the extra characters "於" (in) and "國" (state), we must first segment the term.

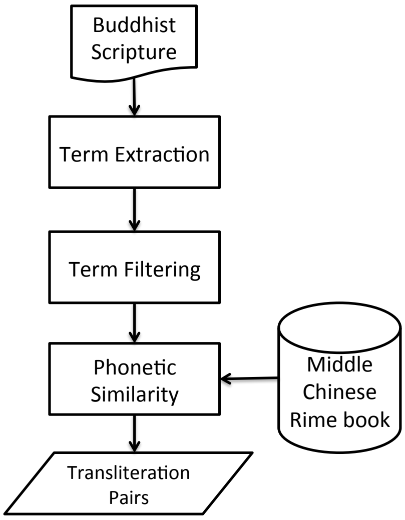

Figure 1: Overview of our system

In order to filter and extract transliteration terms, i we propose a language-model-based filtering method. A language model assigns a probability to a sequence of m words $P(w_1, w_2, \cdots, w_m)$ by means of a probability distribution. The probability of a sequence of m words can be transformed into a conditional probability such as:

$$\begin{aligned} P(w_1, w_2, \cdots, w_m) &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\cdots \\ &\quad P(w_m|w_1, w_2, \cdots, w_{m-1}) \\ &= \prod_{i=1}^{m} P(w_i|w_1, w_2, \cdots, w_{i-1}) \end{aligned}$$

In practice, we can assume the probability of a word only depends on its previous word (bi-gram assumption). Therefore, the probability of a sequence can be approximated as:

$$\begin{aligned} P(w_1, w_2, \cdots, w_m) &= \prod_{i=1}^{m} P(w_i|w_1, w_2, \cdots, w_{i-1}) \\ &\approx \prod_{i=1}^{m} P(w_i|w_{i-1}) \end{aligned}$$

We collect person and location names from the Buddhist Authority Database[1] and also gather Buddhist transliteration terms from The Buddhist Translation Lexicon (翻譯名義集) to create a dataset with 4,301 transliterations for our bi-gram language model.

After building the bi-gram language model, we can apply it to filter the terms from the suffix array method. We formulate transliteration term segmentation into a boundary-checking problem. For each term from the suffix array, we check where the left and right boundaries of the actual

[1] `http://authority.ddbc.edu.tw/`

transliteration are based on the probability of the bi-gram language model.

Continuing with the previous example, “於憍薩羅國” (in Kosala state), we first calculate the probability of the first two characters: $P(\text{於憍}) = P(\text{於})P(\text{憍}|\text{於})$. Then, we calculate the probability of the second and third characters: $P(\text{憍薩}) = P(\text{憍})P(\text{薩}|\text{憍})$. We check the probability of each bi-gram from left to right. If the probability changes sharply from that of the previous bi-gram, the previous bi-gram may be the left boundary of the transliteration. Because the character “於” rarely appears in transliterations, $P(\text{於憍})$ is much lower than $P(\text{憍薩})$. We can thus determine that the left boundary is between the first two characters “於憍”. We also apply a similar way to check the right boundary by calculating the probability of bi-grams from right to left. If the probability of all the bi-grams is low, the term may not contain any transliterations and is filtered out. In practice, we only deal with the leftmost and rightmost three bi-grams to check if the boundary exists because there are few terms which have more than three characters on the left or right-hand side of an actual An English equivalent would be to combine the initial of ‘peek’ /p^{h}iːk/ and the rhyme of ‘cat’ /kæt/ to get ‘pat’ /p^{h} æt/.

2.3 Phonetic similarity

2.3.1 Middle Chinese pronunciation

After extracting the set of loanword candidates, we have to measure the phonetic similarity between every two candidates to get possible loanword pairs. Because the pronunciation of Chinese characters varies diachronically and synchronically, the same Chinese characters may have been pronounced differently in different regions and eras of ancient China. Therefore, we cannot base our measurements of the phonetic similarity of the loanword on modern Chinese pronunciation. We use the rime book Guangyun, compiled during the Northern Song dynasty (960–1127), instead.

Rime books, such as Guangyun, record contemporary character pronunciations with fanqie “反切” analyses. Fanqie represents a character’s pronunciation with another two characters, combining the former’s “initial” and the latter’s “rhyme” and tone. An English equivalent would be to combine the initial of ‘peek’ /p^{h}iːk/ and the rhyme of ‘cat’ /kæt/ to get ‘pat’ /p^{h}æt/.

To employ the Guangyun rime book data in our analyses we must first convert the Chinese characters it uses to represent pronunciation into International Phonetic Alphabet (IPA) notation. We use the reconstruction of middle Chinese pronunciation proposed by Wang Li [Wang 02] for this task. Take the character “洪” for example. Its initial “匣” is converted to IPA [ɣ] and its rhyme “東” is converted to [uŋ], giving us a final reconstructed IPA phonemic form of [ɣuŋ].

2.3.2 Phonetic similarity measurement

We use the ALINE algorithm to measure the similarity between the phonetic forms of every two candidate terms according to their string editing distance. Algorithm 1 shows the pseudocode for the phonetic similarity algorithm which we constructed using dynamic programming. Given x and y strings as input, the algorithm has three string edit actions: *skip*, *substitute*, and *expand*. *Skip* stands for skipping a phoneme in either input string; *substitute* is to substitute a phoneme in one input string with a phoneme in the other input string, and *expand* is to substitute a phoneme in one input string with two phonemes in the other input string. When final consonants or consonants in consonant clusters are dropped in transliterations from Sanskrit to middle Chinese, we have to consider the *expand* action. In Algorithm 1, the functions δ_{skip}, δ_{sub}, δ_{exp} are the scoring functions of the *skip*, *substitute* and *expand* actions.

Algorithm 1 ALINE phonetic similarity algorithm

```
$S[0,0] \leftarrow 0$
$n \leftarrow |x|$
$m \leftarrow |y|$
for $i = 1$ to $n$ do
  $S[i,0] \leftarrow S[i-1,0] + \delta(a_i, -)$
end for
for $j = 1$ to $m$ do
  $S[0,j] \leftarrow S[0,j-1] + \delta(-, b_j)$
end for
for $i = 1$ to $n$ do
  for $j = 1$ to $m$ do
    $S[i,j] \leftarrow \max \begin{pmatrix} S[i-1,j] + \delta_{skip}(x_i) \\ S[i,j-1] + \delta_{skip}(y_i) \\ S[i-1,j-1] + \delta_{sub}(x_i,y_j) \\ S[i-1,j-2] + \delta_{exp}(x_i,y_{j-1}y_j) \\ S[i-2,j-1] + \delta_{exp}(x_{i-1}x_i,y_j) \\ 0 \end{pmatrix}$
  end for
end for
return $S[n,m]$
```

Sound changes in any given language show tendencies [Cambell 04]. For example, the change from dental stop [t] to palatal affricate [tɕ] is common in many languages; however, the dental stop [t] is rarely changed to the glottal fricative [h]. The similarity between [t] and [tɕ] should be higher than the one between [t] and [h]. Therefore, the scoring functions should take the actual phonetic features such as place of articulation, manner, voiced or voiceless into consideration. Table 1 shows a detailed definition of the scoring functions. The constants in the scoring functions are set to $C_{skip} = -10$, $C_{sub} = 35$, $C_{exp} = 45$, and $C_{vwl} = 10$. The phonetic features of each phoneme are represented as vectors, and the function $\text{diff}(p, q, f)$ is the difference value between phoneme p and phoneme q in feature f.

The phonemic features we use are divided into two sets. One is for comparing two vowels, R_V, and one is for comparing two consonants or one consonant and one vowel, R_C. The features in R_V are: *Syllabic*, *Nasal*, *Retroflex*, *High*, *Back*, and *Round*. The features in R_C are: *Syllabic*, *Manner*, *Voice*, *Nasal*, *Retroflex*, *Lateral*, *Aspirated*, and *Place*. The value of each feature is in the range of [0.0, 1.0]. *Place*, *Manner*, *High*, and *Back* features are multi-valued. Table 2

Table 1: Scoring functions

$$
\begin{aligned}
\delta_{skip}(p) &= C_{skip} \\
\delta_{sub}(p,q) &= C_{sub} - \delta(p,q) - V(p) - V(q) \\
\delta_{exp}(p,q_1q_2) &= C_{exp} - \delta(p,q_1) - \delta(p,q_2) - \\
&\quad V(p) - \max(V(q_1),V(q_2)) \\
\text{where} & \\
V(p) &= \begin{cases} 0 & \text{if } p \text{ is a consonant} \\ C_{vwl} & \text{otherwise} \end{cases} \\
\delta(p,q) &= \sum_{f\in R} \text{diff}(p,q,f) \times \text{salience}(f) \\
\text{where} & \\
R &= \begin{cases} R_C & \text{if } p \text{ or } q \text{ is a consonant} \\ R_V & \text{otherwise} \end{cases}
\end{aligned}
$$

shows the values of these four features. The other features only have two values: 0.0 and 1.0. The salience(f) function defines the importance of the feature f. For example, the place of articulation is much more important than aspiration in phonetic similarity comparison, so the salience of the feature *Place* should be higher than that of *Aspirated.* The salience values of the features are listed in table 3.

After computation of algorithm 1, the max value in $S[i,j]$, $0 \le i \le n$, $0 \le j \le m$ is the similarity score of these two phoneme sequences. However, we have to normalize the score because the longer sequence has a higher score. We normalize the similarity score into the range [0, 1] by the following formula: Suppose $AlignScore(x,y) = \max_{0\le i\le n, 0\le j\le m} S[i,j]$, the final similarity

$$Similarity(x,y) = \frac{AlignScore(x,y)}{AlignScore(q,q)}$$

where q is the longest sequence between x and y.

3. Evaluation

3.1 Data set

We choose two Buddhist scriptures as our data set for evaluation from the Chinese Buddhist Canon maintained by Chinese Buddhist Electronic Text Association (CBETA). One sutra we choose is the Samyuktagama (雜阿含經) and the other one is the Abhidharma Mahavibhasa Sastra (阿毘達磨大毘婆沙論). These two sutras are translated into classical Chinese in different era and have different characteristic of transliterations, so they are suitable to evaluate our method.

The Samyuktagama is an early Buddhist scripture which was collected shortly after the Buddha's death. It is the one of the most important sutra in Early Buddhism. An Indian monk, Gunabhadra, translated this sutra into classical Chinese in Liu Song dynasty around 443 C.E. The classical Chinese Samyuktagama has 50 volumes containing about 660 thousands characters.

Table 2: Multi-valued features and their values

Feature name	Phonological term	Value
Place	[bilabial]	1.0
	[labiodental]	0.95
	[dental]	0.9
	[alveolar]	0.85
	[retroflex]	0.8
	[palato-alveolar]	0.75
	[palatal]	0.7
	[velar]	0.6
	[uvular]	0.5
	[pharyngeal]	0.3
	[glottal]	0.1
Manner	[stop]	1.0
	[affricate]	0.9
	[fricative]	0.8
	[approximant]	0.6
	[high vowel]	0.4
	[mid vowel]	0.2
	[low vowel]	0.0
High	[high]	1.0
	[mid]	0.5
	[low]	0.0
Back	[front]	1.0
	[central]	0.5
	[back]	0.0

Table 3: Salience settings of the features

Syllabic	5	Place	40
Voice	10	Nasal	10
Lateral	10	Aspirated	5
High	5	Back	5
Manner	50	Retroflex	10
Round	5		

The Abhidharma Mahavibhasa Sastra, shortly Mahavibhasa, is a vital scripture in Mahayana Buddhism. Its authorship is traditionally attributed to five hundred Arhats around 150 C.E., 600 years after the Buddha's death. The classical Chinese text of the Mahavibhasa was translated by a Chinese monk Xuanzang from the Sanskrit text he got in India at the mid-seventh century in Tang dynasty. Xuanzang was known for his extensive but careful translations of Indian Buddhist texts to Chinese. He took the phonetic corresponding in transliteration very seriously, so the transliterations he did are different from ones translated by previous translators. The classical Chinese text of the Mahavibhasa is immense. It has 200 volumes and more than 1.63 billion characters.

3.2 Experimental results

Because the texts of the two scriptures in our data set have billion characters, it is not plausible to get all the transliteration pairs as the ground truth to evaluate our method. Therefore, we collect the transliteration terms in the extraction result of the suffix array method from the Samyuktagama and the transliteration terms recorded ap-

Table 4: Evaluation result

	Recall	Precision
all transliteration pairs	0.7500	0.6206
distinct transliteartion pairs	0.6667	0.4942

pearing in the Samyuktagama from the Buddhist Authority Database maintained by Dharma Drum Buddhist College. Then, we check the terms from the Samyuktagama have the other transliteration variations which appear in the Mahavibhasa or not. We search theses transliteration terms in the Fo Guang Buddhism Dictionary and Ding Fubao's Dictionary of Buddhist Studies to get all the transliteration variations of the term and then check if some of them appearing in the Mahavibhasa. If so, we add this two terms as a transliteration pair into our ground truth.

Table 4 shows the experimental results of our method. The measurements we use are Recall and Average Precision. Because some transliterations are identical in the Samyuktagama and the Mahavibhasa, we also construct a configuration excluding these identical ones in our ground truth. The evaluation results show that our method can extract almost 75% transliteration pairs and the average precision achieves 62%. It shows our method is effective to extract the transliteration pairs in Buddhist scriptures.

4. Discussion

4.1 Effectiveness of transliteration pair extraction

Our method can extract many transliteration pairs which transliterated from the same Sanskrit words in the Samyuktagama and the Mahavibhasa such as "比丘" and "苾芻" (Bhikhu, male Buddhist monks), "目犍連" and "目乾連" (Maudgalyayana, one of the Buddha's closest disciples), "波羅提木叉" and "般羅底木叉" (Pratimoksha, Buddhist moral discipline). The transliteration pairs whose pronunciations are not similar in modern Chinese can also be extracted by our method like "優婆塞" and "鄔婆索迦" (Upasaka, male followers of the Buddhism who are not monks), "迦毘羅衛" and "劫比羅" (Kapilavastu, the name of an ancient kingdom where the Buddha was born and grew up), "迦旃延" and "迦多衍那" (Katyayana, one of the Buddha's closest disciples). Our method can find the person names, location names, and Buddhism specific terms that take important roles in historical Chinese phonology.

Our method also draws out some terms that have similar pronunciations in Sanskrit. For example, our method find the transliteration pair "三昧" and "三摩地" for the Sanskrit term "Samadhi" (mental concentration). Besides, the term "奢摩他" (Samatha, calm abiding) that has the similar pronunciation is also extracted by our method. Samatha is a mediation practice to achieve Samadhi. These terms may help the research of Buddhism and phonology.

In addition, we discover transliterations also vary even in the same scripture through our method. In the Samyuktagama, the Sanskrit term "Magadha" (the name of an ancient Indian kingdom) has three different transliterations: "摩竭陀", "摩竭提", and "摩伽陀". The term "Arhat" (a spiritual practitioner who has realized certain high stages of attainment) also has three transliterations: "阿羅漢", "阿羅訶", and "阿羅呵". These variations may help the study of historical Chinese phonology and philology. It also shows the importance that the terms' actual pronunciations should be considered while dealing with the Buddhist scriptures.

4.2 Error cases

Although our method can extract and identify most transliteration pairs, there are some transliteration pairs cannot be identified. The error cases can be divided into several categories. The first one is the transliterations are not extracted by the suffix array method. For example, the transliteration term "末土羅" (Mathura, an ancient city in northern India) is not extracted by the suffix array method because it appears only once in the Mahavibhasa. It is the limitation of the suffix array method.

The other case is that rarely-used characters that are not in our data set for constructing the bi-gram language model. Take the transliteration "阿練若" (Aranya, the forest) in the Mahavibhasa for example. The middle character "練" is seldom used in transliterations and not covered in our dataset. The widely-used transliteration for the Sanskrit "Aranya" is "阿蘭若". Therefore, our method filters out these unseen characters to drop out the transliteration terms.

The final case is the phonemic problem of the Chinese transliteration itself. For example, the Sanskrit term "Upasika" (female followers of the Buddhism) are transliterated into "優婆夷" in the Samyuktagama and "鄔波斯迦" in the Mahavibhasa. The pronunciation of the last character "夷" in the transliteration "優婆夷" in the middle Chinese is [ji]. The semi-vower [j] is not similar to the dental fricative [s] in Sanskrit origin. Thus, the phonetic similarity score is too low to identify by our method.

5. Conclusion

The identification of transliterations of foreign loanwords is an important task in research fields such as historical linguistics and digital humanities. However, the amount of the historical literature is so immense. Therefore, we propose an approach which can identify transliteration pairs automatically in classical Chinese texts. Our approach comprises two stages: transliteration extraction and transliteration pair identification. To extract more possible transliterations, we adopt a hybrid method consisting of a suffix-array-based extraction step and a language-model based filtering process. Next, we compare the extracted transliteration candidates for phonetic similarity based on their pronunciations in the middle Chinese rime book Guangyun with ALINE algorithm. Pairs with similarity above a certain threshold are considered transliteration pairs. To evaluate our method, we constructed an evaluation set from the two Buddhist texts such as the Samyuktagama and the Mahavibhasa, which were translated into Chinese in different eras. The recall of our method achieves 0.75 and the precision is 0.6206. The results show our method is effective to extract the transliteration pairs in classical Chinese texts. Our method can find the relationship of sound correspondence among the immense classical literatures to help many researches such as historical linguistics and philology.

References

[Cambell 04] Cambell, L.: *Historical linguistics: an introduction*, The MIT Press (2004)

[Covington 96] Covington, M.: An algorithm to align words for historical comparison, *Computational Linguistics*, Vol. 22, No. 4, pp. 481–496 (1996)

[Kondrak 03] Kondrak, G.: Phonetic alignment and similarity, *Computers and the Humanities*, Vol. 37, No. 3, pp. 273–291 (2003)

[Mackay 05] Mackay, W. and Kondrak, G.: Computing word similarity and identifying cognates with Pair Hidden Markov Models, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pp. 40–47 (2005)

[Manzini 04] Manzini, G. and Ferragina, P.: Engineering a lightweight suffix array construction algorithm, *Algorithmica*, Vol. 40, No. 1, pp. 33–50 (2004)

[Ristad 98] Ristad, E. and Yianilos, P.: Learning string-edit distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 522–532 (1998)

[Wang 02] Wang, L.: *Historical Chinese Phonoloy*, Zhonghua Book Company (2002)